Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering

Master's Thesis

Gallery Website development using .NET Core and TypeScript

Author: Bc. Vladimir Zinovev

Supervisor: Ing. Jiří Brožek, Ph.D.

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Bc. Vladimir Zinovev

Systems Engineering and Informatics

Informatics

Thesis title

Gallery Website development using .NET Core and TypeScript

Objectives of thesis

The main goal of the diploma thesis is to develop the SPA and MPA Gallery Website versions using .NET Framework, Angular.js, and Vue.js. The main aim of the web application is to arrange the collections of user photos and present the images in different ways.

Methodology

The thesis consists of the theoretical and practical part.

The theoretical part of the thesis will be based on the study and review of available information sources. The knowledge gained during this process will be used as a basis for the practical part.

The practical part concerns the design and implementation of a photo gallery web application. The application will consist of a back-end and front-end part. The back-end will be implemented using .NET Core framework and will offer API that will be consumed by the front-end part of the application. Front-end of the application will include features for displaying uploaded images of users to present the uploaded photos, arrange them in different ways, scroll, edit, commend user's photos, write comments, etc. Angular.js and Vue.js would be overviewed and implemented for the web API presentation layer with other tools like Bootstrap, jQuery, Okta, Babel, etc.

The application will be deployed, tested and the experience and feedback from the development, testing and using will be evaluated.

The proposed extent of the thesis

60-80 pages

Keywords

C#, .NET Core, TypeScript, JavaScript, Bootstrap, Angular.js, Vue.js

Recommended information sources.

HANCHETT, Erik, LISTWON, Benjamin. 2019. Vue.js in Action. Manning Publications. 304 p. ISBN 978-1617294624
HORSTMANN, Cay S. 2020. Modern JavaScript for the Impatient. Pearson Education, Inc. 288 p. ISBN 978-0-13-650214-2
https://angular.io/
https://developer.mozilla.org/
https://docs.microsoft.com/
https://vuejs.org/
https://www.typescriptlang.org/
https://www.w3schools.com/
CHERNY, Boris. 2021. Programming TypeScript: Making Your JavaScript Applications Scale. O'Reilly Media. 352 p. ISBN 978-1492037651

Expected date of thesis defence

2021/22 SS – FEM

The Diploma Thesis Supervisor

Ing. Jiří Brožek, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 1. 11. 2021

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 23. 11. 2021

Ing. Martin Pelikán, Ph.D.

Dean

In Prague on 31. 03. 2023

Declaration

The author of "Gallery Website development using .NET Core and TypeScript" and its author, asserts that it is created by himself only. The author declares the diploma thesis lacks any knowledge that have been previously divulged elsewhere. Its sources are listed in the References.

In Prague on 30th of March, ____________________________ Vladimir Zinovev

Acknowledgement

I would like to kindly thank Ing. Jiří Brožek, Ph.D. for advice and consultations that were helpful for me and my work on the diploma thesis.

Gallery Website development using .NET Core and TypeScript

Abstract:

The main goal of the diploma thesis is to develop a single-page application (SPA) of gallery website. The multiple-pages application (MPA) is discussed for the comparison of both approaches. The main aim of the web application which is called gallery website is to arrange the collections of photos and present them in an interactive way.

The topic for the diploma thesis was chosen by me in the year 2021. The project assignment has been created and approved around a year ago. Since that time many changes appeared in the technology stack I would like to use for the practical part. For example, .NET is no longer called .NET Core in the year 2023. The name of the diploma thesis remains the same as it was approved in the year 2021.

While working on the thesis, I tried out several possible technologies for the website development and finally during my summer vacation my technology stack has been formed. I started to develop in Axios, Nodejs, Vue.js, ECMAScript, and so on. So that is why I used primarly this technologies for the practical part of the thesis.

I started to develop in JavaScript. I developed a SPA based on Vue.js framework. For this task it is possible to use Nodejs and npm as for the executive runtime environment.

The final application has an interactive photo grid. It is possible to rotate, move, resize, rearrange photos in different ways. A user could see the available options for a photo presentation simply by clicking on a photo. The project's client interface allows to reshape the images as well.

Keywords: JavaScript, Nodejs, Vue.js, Axios, SPA

Vývoj Webové galerie pomocí .NET Core a TypeScript

Abstrakt:

Cílem této diplomové práce je vývoj jednostránkové webové aplikace (SPA), která se nazývá web galerie. Pro srovnání s SPA je diskutována i vícestránková aplikace (MPA). Hlavním cílem webové aplikace galerii je uspořádaní sbírky fotografií a zároveň jejich prezentovaní interaktivním způsobem.

Tématu diplomové práce jsem si vybral v roce 2021. Zadání projektu bylo vytvořeno a schváleno zhruba před rokem. Od té doby se v technologickém světě se objevilo mnoho změn, které byli využité pro praktickou část projektu. Například .NET se nejmenuje .NET Core v roce 2023. Název diplomové práce však zůstává stejný, jaký byl i v roce 2021.

Při zpracování projektu jsem vyzkoušel několik možných technologií pro vývoj webových stranek a nakonec se mi během letních prázdnin vytvořil můj technologický stack. Začal jsem vyvíjet v Axios, Nodejs, Vue.js, ECMAScript, atd. Pro praktickou část jsem použil tyto technologie.

Začal jsem vyvíjet v JavaScriptu vytvořit SPA, které je založené na frameworku Vue. Pro tuto úlohu je možné použít Nodejs a npm jako pro výkonné runtime prostředí.

Finální aplikace má interaktivní fotografickou mřížku. Fotografie je možné různými způsoby otáčet, přesouvat, měnit jejich velikost, přeskupovat je. Uživatel může jen kliknutím na fotografii vidět dostupné možnosti prezentace této fotografií. Klientské rozhraní projektu také umožňuje přetvářet obrázky.

Klíčová slova: JavaScript, Nodejs, Vue.js, Axios, SPA

1. Table of content

2. Introduction

The diploma thesis assignment was created in the year 2021. Since that time my technology stack changed. The web technologies changed as well, too.

The diploma thesis is about the development of gallery website.

Gallery website represents an online web application that has a photo grid with the option for image manipulation. The website has a single page as a layout. Essentially the project is produced in JavaScript and its programming capabilities. They would be covered in objectives and methodology part.

The literature review represents the theoretical part of the project. It is the study of books, online web sources, and other materials. Some milestone technologies are explained in depth with their current stable versions, sponsors, etc.

The project is about full-stack development of a software product. There are many analogue technologies are shortly discussed as the examples of the web development approaches. The resemblance and difference of analogue technologies and core root technologies are the parts of the discussion and analysis.

In the literature review there are also the description of basis web technologies that are necessarily used in each software development process.

In practical part are discussed the used information technologies and their handling and treatment in project are explained. There are some code examples and images for better understanding of the application. Inner project's logics is shown as the consistent part of the thesis.

Results and discussion sum up the thesis output and appraise the given goals from objectives and methodology part. The web application components are represented as figures for the project's preview.

In the conclusion the further research opportunities are given as the result of project exploration.

The thesis also seizes references, list of abbreviations, and the code sample. These parts make the thesis polished and informative.

The essential part of the diploma thesis is the created gallery website. It is attached to the thesis as the compressed file.

3. Objectives and Methodology

3.1 Objectives

„Today’s software and systems engineers are facing an increasing number of challenges as they attempt to develop new products and systems faster, with higher quality and rich feature content.“ (Broy, Simmons, 2009)

The goal is to create a modern and useful application using Vue framework and some attendant technologies that are listed below in methodology.

The main aim is to create an application based on TypeScript frameworks. The Vue.js framework is the core framework for the practical part. The development in the Vue.js framework is compared with the Angular.js framework implementations in literature review.

In the thesis the single-page application of the gallery website is presented. The thesis analyses the technological stack for web development. So it is about software development and the information technologies (IT) sector.

There are different pieces of IT sector. In the thesis I would focus only on web development and its software development lifecycle.

Gallery website project is close to the commonly named full-stack development. The full-stack development is the main objective of the current diploma thesis.

The full-stack development is a comprehensive way to do programming and create applications. The full-stack developer should be a proficient expert in almost the all fields of a chosen software segment.

End-to-end application software development is also called full-stack development. The development of the SPA gallery website is a full-stack development. The final web application project consists of a client and server, interfaces, logic, and program’s workflows. The goal is to present all the parts of gallery website.

The client side or front-end refers to the user interfaces while the server side or back-end refers to application’s workflows and robust logic. Both of the software application parts should be user friendly, consistent, and robust.

Gallery website project advices how to build fully robust web applications as a full-stack software development process.

„As more features are being defined for a product or system, the discipline of requirements engineering has increased in importance to help manage the development of the features throughout the product life cycle.“ (Broy, Simmons, 2009)

All parts of the diploma thesis are explained informatively using the referenced sources. So everything mentioned above is the objectives of the current diploma thesis.

3.2 Methodology

During my work on the thesis, I tried out several possibilities for the back-end and front-end sides of the project. decided to emphasize Nodejs, JavaScript, TypeScript and Vue.js framework. It is possible to use Nodejs and its package manager called Node Package Manager (NPM) for the task.

The practical part is the gallery website project based on Vue framework. It is pronounced as [vjuː], similar to English word ‚view‘. It is a quite famous back-end framework and it is used for the creation of modern user interfaces and application programming interfaces. Vue is a progressive backend framework and ecosystem at the same time.

The framework Vue.js or simply Vue is complex and consists of several parts. They are HTML5, CSS3, and ECMAScript (similar to JavaScript (JS)). Vue provides a developer with a component-based and declarative programming model, functions, and paradigm. All this makes the development of simple and complex user interfaces clear and easy.

Visual Studio is the integrated development environment (IDE) for .NET framework. It is free to install the Community edition of Visual Studio. There are pre-paid Enterprise and Professional editions[1].

TypeScript is the most significant thesis technology. This is the essential programming language for Vue.js or Angular.js developer.

The progressive Vue framework is discussed below in the thesis. I would use Vue.js[2] framework for the best user experience with my web application.

Other discussed technologies are TypeScript, Angular.js, .NET, git, software testing approaches, Nodejs, in the case of RESTful architecture. This technologies are assumed, analyzed and discussed in literature review, practical part and conclusion for the example and the comparison.

[1] Visual Studio download page [https://visualstudio.microsoft.com/]

[2] Vue.js documentation [https://vuejs.org/]

All the discussed frameworks have been compared with Vue.js framework in the diploma thesis. The frameworks represnt different way of thinking so their features are assumed and studied.

The used IT technologies are observed for the similarity and resemblance. This makes point for further researches.

4. Literature review

4.1 Software development lifecycle and REST

Software development lifecycle (SDLC)[3] expresses steps that are involved in software creation. They are:

- requirements analysis and initiation
- planning and prototyping
- software and architecture design
- development and implementation
- testing and verification
- deployment
- maintenance and hypercare

SDLC methodology comes from systems development methodology of systems development lifecycle. The systems development life cycle consists of:

- planning
- analysis
- design
- implementation
- maintenance

SDLC presents in every software development. That means each program goes through the stages of SDLC.

In web development the SDLC steps are the leverage points as well. The software development lifecycle is being considered together with Representational State

[3] Software development lifecycle [https://pmbasics101.com/software-development-life-cycle/]

Transfer (REST) principles[4]. It is an architectural style for distributed hypermedia systems like online gallery, forum, chatbot, e-commerce, and the others.

The REST principles are the following:

- separation of concerns
- specifications for cache
- hierarchy
- explanatory instructions
- principle of generality

A web application is an application that is run and accessed in a web browser like Chrome, Edge, Firefox etc. The running program is started and fully controlled by a programmer. In this way a web application is stored on a remote server and users access the client interface via a web browser. The request and response pipeline is being maintained by a software developer. The Hypertext transfer Protocol (HTTP) is used for the communication with a server. HTTP and representational state transfer (REST) application programming interface (API) are commonly named RESTful APIs.

[4] Representational State Transfer [https://restfulapi.net/]

4.2 Interfaces overview

User interface[5] (UI) in our case are represented by the clients interfaces (machines interfaces) of a web application programming interface in computer science that allows users to make online changes on a website they visit. It is the point for communication and human-computer interaction an a web page, application, device, etc. Screens, displays, mouses, keyboards, touchpads and so on are the parts of a user interface. User interface represents what a user would see during the interaction with a machine.

User interfaces could be done for web pages, devices, aplications and so on. The user interface has different spaces, structures, and other elements on a display where users collaborate with its various and different elements like online forms, buttons, and other components. Elements include input controls, navigational and informational, containers, flexboxes. There could be some other types of the elements.

Machine interfaces are called client interfaces. User agent for a web is a machine that is called client. It uses application programming interfacers, message passings, remote method invocations (RMI), and the other techniques[6].

Client is a tool like a machine or a programm that allows users (persons) to deal with the APIs and server's remote functionality. Client makes call to a server every time a user wants to access a server's functionality. The information is then displayed and rendered by a client as well. Every user has own login credentials and many users could log onto a client at the same time. Users start their sessions on a client.

The client or user interface depicts the displayed information for the end users. It is a program that controls the user's display and allows interactions with the server through its client for calculations, data manipulation, program's logic and so on.

A good user interface enables the best user experience (UX). User interface is focused on the functionality and the design of a finished product (a web application in

[5] User interface [https://www.usability.gov/what-and-why/user-interface-design.html]

[6] Remote method invocation [https://www.theserverside.com/definition/Remote-Method-Invocation-RMI]

my case) while the user experience is focused on ideation, management, and delivery of a project. The appeal of a web application should be consistent, continuous, aesthetic, simple, and sustainable for the end users.

The progressive Vue framework is discussed below in the thesis. I would use Vue.js[7] framework for the best user experience with my web application.

Other discussed technologies are TypeScript, Angular.js, .NET, git, software testing approaches, Nodejs, RESTful architecture. This technologies are assumed, analyzed and discussed in literature review, practical part and conclusion for the example and the comparison. The technologies are compared with Vue.js framework way of thinking. The technologies are observed for similarity and resemblance.

[7] Vue.js official page [https://vuejs.org/]

4.3 JavaScript

JavaScript is also called ECMAScript. It was developed by Ecma international[8]. The scripting programming language is commonly used for server side.

JavaScript has many versions. Its first edition came out in the year 1997. Guy L. Steele Jr. was the founder of JavaScript.

ECMAScript (ES2015) emerged eighteen years later in 2015. Today we use the 13th edition of ECMAScript (ES2022).

The programming language is universally used for front-end components and user-client interactions. Web forms validating, cache storing, calculation logics – this are the main purposes to use javaScript in the front-end.

The files written in JavaScript have the etension „.js". They are placed inside the <script> tag of a HTML layout.

Variable declarations of var, let, const are common for JS. Undeclared variables are specific for JS. Variable declared using var keyword is a globally scoped variable, those variable which have let keyword are blok scoped. Const declared variable cannot be changed and they are accessed only in a block they are declared.

Let enables impicit conversion of a value it stores. In this case, the converted alphanumeric value preserves itself in any order of its conversion. No data lass occurs while the implicit conversions. Automatic type conversion and type casting are the other name for implicit converting. JS is suitble for storing large amount of data from client interface.

This programming language has many frameworks for comfortable and smooth development.

The application of JS is various. Some of them are:

- server APIs

[8] JavaScript origins [https://www.ecma-international.org/]

- mobile APIs
- single-page applications
- web APIs
- game development
- smartwatches
- artificial intelligence

4.4 Vue.js implementation

It is approachable due to its intuitive application programming interface (API). It is built on top of standard HTML, CSS, and JavaScript. It has a well-written documentation. I used the documentation from the official Vue.js website.

Vue.js is performant because it is an optimized system with a good build-in compiler. Vue is truly reactive and requires minimum manual optimization.

The Vue framework has many sponsors. They are listed on the official Vue.js web page.

Vehikl is a software consultancy that has its specialization primarily in PHP (originally Personal Home Page, now stands for PHP: Hypertext Preprocessor) and JavaScript. Vehikl creates the leading web applications in many frameworks such as Vue, React, Laravel, Express.js, Nest.js. There are many sponsors and teams who work with Vue approach. This framework is highly demanded nowadays.

Due to its increasing popularity, there are many resources to learn Vue. Primarily, they are free and open-sourced.

The most popular tutorial resource to learn Vue.js is the website LearnVue[9].

In the year 2023 the Vue.js version 2 would be deprecated on the 31st of December. So, the main version is Vue 2 amd Vue 3. I would focus on these versions in my thesis.

There are a wide range of sources to learn Vue.js. The official Vue.js website is one of the best online tutorial platforms of the kind. Everyone would be satisfied after visiting this website. I value this a lot due to its broad, informative and clearly written documentation.

[9] Learn Vue tutorials [https://learnvue.co/]

The documentation[10][11] on Vue and the examples[12] helped me to create my first web application using this framework.

[10] Vue 2 documentation [https://v2.vuejs.org/]
[11] Vue 2 to Vue 3 migration documentation [https://v3-migration.vuejs.org/]
[12] Vue 2 examples (‚Hello World!‘ example) [https://vuejs.org/examples/#hello-world]

4.5 Purposes to use Vue

The Vue framework could be used for the following purposes:

- Single-Page Applications (SPA)
- Embedded web component on a web page
- Single-file components
- Composition API
- Options API
- Static HTML enhancing without a build phase
- Server-Side Rendering or Fullstack (SSR)
- Static Site Generation or Jamstack (SSG)
- Terminal applications
- Mobile applications
- Desktop applications
- Web Graphics Library (WebGL)

The approaches to use Vue.js could be different. In a time, there would be more and more of software examples and applications.

In Vue 2, there exist only one way to implement components that is called Options API. In Vue 3 there is new way to create components called the Composition API. As for the year 2023, there exist two possibilities to create components in the framework[13].

[13] Vue components implementations https://fjolt.com/article/vue-composition-api-vs-options-api

4.6 Options API and Compositions API

The key difference in Options API and Composition API is their syntax. The Options syntax is more complicated and outdated. For example, there is a counter component created in Options API:

```
<h1> </h1>

<button @click="incrCounter”> Click Me

</button>

<script>

export default {data () {return { counter: 0 } },

methods: {

incrCounter: function() { this.counter += 1; } } }

</script>
```

The example of same component created in Composition API is presented below:

```
<h1></h1>

<button @click="incrCounter"></button>

<script setup>

import {ref} from 'vue'
```

```
let counter = ref (0);

const incrCounter = function () {counter.value += 1; }

</script>
```

The Composition API syntax is more elegant and shorter than the same things inside Options API. The key differences are the followinf:

In the case of Composition API we use so called reactive variables – the ref statement is used for this

Ref returns an entire object – so we do not use an entire prototype in the case of the Composition API, but simply increment the counter without calling an object. In case of Options API, we use JavaScript object prototypes[14]. So, we call a method object and use it for the counter. Methods are objects in JavaScript.

[14] Object prototypes in JavaScript [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object_prototypes]

4.7 Angular.js and Vue.js differences

Vue.js and Angular.js are both web application frameworks.

Vue.js is a framework that is ideal for building single-page applications. On the other hand, Angular.js is opinionated and mandates that applications should have a certain structure.

In comparison to Vue JS, Angular JS is implemented solely using the JavaScript language, whereas Vue JS leverages an HTML-based template syntax and constructs utilizing the principles of Models and Components.

Angular.js is more complex and suites for large projects like multiple-page applications then Vue.js[15].

Vue JS can provide a more simplified approach to product development, which may be preferred by experienced developers. Particularly for those lacking strong JavaScript skills, Vue JS is a valuable option to consider.

For those working with the Laravel community, Vue JS should be the primary framework choice as it is highly regarded and can lower processing time by 50% while freeing up server space.

Angular JS is the preferred alternative for programmers who need to work with both client-side and server-side systems. In case of a desire to develop complicated web applications with numerous components and demanding requirements, Angular JS would be an excellent option.

It is also a feasible option if it is required to select new elements for an existing project.

Vue JS is considered lightweight and functions similarly to a slimmer version of Angular JS, making it a preferred choice over Angular JS for those who desire a less

[15] Angular.js and Vue.js comparison [https://kinsta.com/blog/angular-vs-vue/]

directive framework. On the other hand, Angular JS is favored for building intricate applications with good quality.

Determining which framework is perfectly suited comes down to the developer's learning capabilities and the nature of the project.

In a nutshell, both Angular JS and Vue JS, which are both JavaScript frameworks, have the potential to produce excellent results in application development.

4.8 Git versioning

Git and Gitbash are widely used for the software development. They could be simply installed via this link[16]. Besides, everybody knows github.com or gitlab.com. On the websites we could store the code or use the sandbox templates to learn coding.

The git and gitbash technologies are free and open source distributed version control systems. They are designed to handle everything about the code for any project. Just download Git and then you can pull your repositories to an online cloud-based Git repository.

Git is a technology used for versioning and maintaining the code. Knowing Git is compuilsory for every developer and programmer. This technology is the part of each job interview.

```
git clone https://github.com/url-to-repo-here

    /* clone the remote branch into the workstation */

git status

    /* Output is: On branch master

      Your branch is up-to-date with 'origin/master'.

      Changes not staged for commit:

     (use "git add <file>..." to update what will be committed)

     (use "git checkout -- <file>..." to discard changes in working        directory)

     modified:   README.md

     no changes added to commit (use "git add" and/or "git commit - a") */
```

[16] Downloads of git and gitbash for different platforms [https://git-scm.com/downloads]

```
git add .

        /* all the changed files now are the staged files */

git commit -m "comment”

        /* commit the staged files and write a comment */

git push

        /* recent commit transmits to a remote branch */
```

The git commands above push staged files on our local branch to a remote branch. That is how git branch grows. Using git, it is possible to manage source code. Git is a DevOps tool.

Git, gitbash, git extensions, gitlab and other alternatives could be used for software versioning.

4.9 Software testing

The role of testing is to validate life cycle artifacts of a software product that are generated by customers, software engineers, requirements engineers or designers. Testing could find out the defects in requirements.

„Software testing is the process of executing software with the intent of finding errors [Myers 1979], and it basically supports validation & verification (V&V) activities."

Nowadays the primary model for a software testing is called the „V" model [VModel-XT 2008, V-model]. There is also a commonly named „W" model [W-model]. The last model was developed and matured by Paul Herzlich in 1993.

The W-model originates from its precessor, the V-model. It is a more comprehensive variation of the „V" model. The „W" model has more steps and phases while testing then „V" model. It is preferably to use „W" model due to its complex and broad measurment of code. W-model tests coverage is larger then in V-model.

The testing models are interconnected. W-model adresses V-model and vice versa. There are several verification and validation tests approaches.

We have a system under test (SUT) and the test executors perform test manually against the SUT.

It is possible to imlement the automatic generation of test cases that could be accomplished using model-based testing (MBT) and then the generated tested could be repeatively fastly executed by a tester.

The automated tests could be improved and recoded according to the requirements. This is called requirements-driven system testing.

The three types of models for model-based software tesing are the following:

- requirements models that test the designed and expected behaviour of the software product
- models created directly from the source code
- usage models follow the potential user’s actions

4.10 .NET framework

The .NET Framework[17] (it is also sometimes called the .NET Core or simply „dot net“) is among the most popular open-source computer software framework nowadays. .NET was previously named .NET Core.

This framework is the core component for Windows web-based software development, data science, machine learning, game development, database management, etc.

The .NET usually runs on Windows operating system (OS). There are versions for Linux and Mac OS. The current stable release is 7.0.4 according to the official Microsoft website[18].

[17] .NET framework official website [dotnet.microsoft.com]

[18] .NET stable release [https://dotnet.microsoft.com/en-us/download/dotnet/7.0]

5. Practical part

5.1 Vue and Vite

Vue.js installation is quite simple. The terminal could be used for that. It i salso possible to install Vue via Node.js. I used my this terminal commands for the installation:

```
npm install vue
```

```
npm install -g @vue/cli
```

The above commands are the very first two steps to start development in Vue.

Firstly, the Vue framework is installed itself. So does npm install vue command

Secondly, the latest Vue-related packages are added. All packages are upgraded after running npm install -g @vue/cli command.

5.2 Create Vue project

The following command is used to start the template application:

```
npm init vue@latest
```

The last command above installs the official Vue scaffolding tool that is called create-vue. The terminal window would display the following output:

```
C:\VueProjects> npm init vue@latest

        Need to install the following packages:

  create-vue@3.6.1

Ok to proceed? (y) y
```

```
Vue.js - The Progressive JavaScript Framework

√ Project name: ... vue-project1

√ Add TypeScript? ... No / Yes

√ Add JSX Support? ... No / Yes

√ Add Vue Router for Single Page Application development? ... No / Yes

√ Add Pinia for state management? ... No / Yes

√ Add Vitest for Unit Testing? ... No / Yes

√ Add an End-to-End Testing Solution? » Playwright

√ Add ESLint for code quality? ... No / Yes

√ Add Prettier for code formatting? ... No / Yes
```

5.3 Start server

```
Scaffolding project in C:\VueProjects\vue-project1...

Done. Now run:

  cd vue-project1

  npm install

  npm run format

  npm run dev

C:\VueProjects>cd vue-project1

C:\VueProjects\vue-project1> npm install
```

```
npm WARN deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead

added 474 packages, and audited 475 packages in 25 s

111 packages are looking for funding

  run `npm fund` for details

found 0 vulnerabilities

C:\VueProjects\vue-project1> npm run format

> vue-project1@0.0.0 format

> prettier --write src/

src\App.vue 179ms

src\assets\base.css 11ms

src\assets\main.css 5ms

src\components\__tests__\HelloWorld.spec.ts 14ms

src\components\HelloWorld.vue 45ms

src\components\icons\IconCommunity.vue 5ms

src\components\icons\IconDocumentation.vue 3ms

src\components\icons\IconEcosystem.vue 3ms

src\components\icons\IconSupport.vue 3ms

src\components\icons\IconTooling.vue 4ms

src\components\TheWelcome.vue 23ms
```

```
src\components\WelcomeItem.vue 12ms

src\main.ts 8ms

src\router\index.ts 8ms

src\stores\counter.ts 7ms

src\views\AboutView.vue 4ms

src\views\HomeView.vue 4ms

C:\VueProjects\vue-project1> npm run dev

> vue-project1@0.0.0 dev

> vite

  VITE v4.2.0 ready in 493 ms

  ➜ Local: http://127.0.0.1:5173/

  ➜ Network: use --host to expose

  ➜ press h to show help
```

5.4 Vue.js project discussion

The scaffolded created project on the workstation is a single-page application (SPA) based on Vue 3 backend and Vite build setup. Now it is possible to insert and use Vue Single-File Components (SFCs). They would be discussed further in the thesis. Node.js and npm are used as the project's server side.

Npm run command sets the node environment variable to the node executable and after that the npm is started and executed. Npm run requires project to have a node_modules directory. If the node_modules folder doesn't exist, i tis necessary to create it so the command npm install is used for that.

Npm run format runs code formatting and prettier that has been added in the project configuration on its scaffolding stage. Prettier is… Npm run format also overwrites the source files in the project's directory.

The project is started in the browser by using the command npm run dev in the same terminal window after scaffolding and formatting.

The example project uses TypeScript instead of simple JavaScript together with ESLint for the code accuracy.

This kind of Vue projects has the size of 187 MB after its creation on the workstation's hard drive. It is quite a large size of a project on a hard drive. The project's size is big because of many used features and due to its discussed configuration that includes TypeScript. testing, code management, routing, and so on.

In comparison with the previous example the GallerySPA project has much smaller size. In comparison to the previous example it is much less because GallerySPA uses Axios and contains no tests inside it. Axios handles requests and responses of data retrieved back to the web application.

6. Results and Conclusion

The main goals of the thesis were reached. The thesis practical part is the robust single-page application.

The development of a single page application for a gallery website was quite comprehensive and difficult. The main purpose of the online album web application has been reached.

The final application has an interactive photo grid. You can rotate, move, resize, rearrange your photos and much more. Users can simply click on the photo to see the options available for presenting the photo. You can also refactor images using project's client interface.

The main aim was to create a gallery website and organize it to present any photo collection in an interactive way.

7. References

1. CANTELON, Mike, HARTER Marc, HOLOWAYCHUK, T.j., RAJLICH, Nathan. 2014. Node.js in Action. USA, New York: Manning Shelter Islands publications Co. 396 p. ISBN: 978-1-61-729057-2.

2. FOGUS, Michael. 2013. Functional JavaScript. USA, Sebastopol: O'Reilly Media, Inc. 240 p. ISBN: 978-1-449-36072-6.

3. PARKER, Daniel. JavaScript with Promises. Managing asynchronous code. USA, Sebastopol: O'Reilly Media, Inc. 88 p. ISBN: 978-1-449-37321-4.

4. WILSON, Chris. 2014. RaphaëlJS. Graphics and Visualization on the Web. USA, Sebastopol: O'Reilly, Media Inc. 114 p. ISBN: 978-1-449-36536-3.

5. CROWDER, T.J.. 2020. JavaScript: The New Toys. Canada: Willey. 608 p. ISBN: 978-1-11-936797-6.

6. CECCO, Raphaele. 2011. Supercharged JavaScript Graphics. USA, Sebastopol: O'Reilly Media, Inc. 282 p. ISBN 978-1-449-39363-2.

7. HANCHETT, Erik, LISTWON, Benjamin. 2019. Vue.js in Action. USA, New York: Manning Shelter Islands Publications Co. 304 p. ISBN 978-1-61-729462-4.

8. HORSTMANN, Cay S. 2020. Modern JavaScript for the Impatient. Pearson Education, Inc. 288 p. ISBN 978-0-13-650214-2.

9. CHERNY, Boris. 2021. Programming TypeScript: Making Your JavaScript Applications Scale. O'Reilly Media, Inc. 352 p. ISBN 978-1-492-03765-1.

10. CRAVENS, Jesse, JEFF, Burtoft. 2013. HTML5 Hacks. USA, Sebastopol: O'Reilly Media, Inc. 488 p. ISBN: 978-1-449-33499-4.

11. CROWTHER, Rob, LENNON, Joe, BLUE, Ash, WANISH, GREG. HEILMANN, Christian. 2014. HTML5 in Action. USA, New York: Manning Shelter Islands Publications Co. 444 p. ISBN: 978-1-61-729049-7.

12. MURPHY, Christopher, CLARK, Richard, STUDHOLME, Oliver, MANIAN, Divya. 2012. Beginning HTML5 and CSS3. The Web Evolved. Next generations web standards. USA, New York: Apress. 604 p. ISBN: 978-1-4302-2874-5.

13. MEYER A., Eric. 2004. Cascading Style Sheets. Visual presentation for the web. The Definitive Guide. USA, Sebastopol: O'Reilly Media, Inc. 510 p. ISBN: 0-596-00525-3.

14. BERNHARDT, Manuel, ROPER, James. 2016. Reactive Web Applications. Covers Play, Akka, and Reative Streams. USA, New York: Manning Shelter Islands Publications Co. 304 p. ISBN: 978-1-63-343009-9.

15. FAIN, Yakov, MOISEEV, Anton. 2019. Angular Development with TypeScript. Second Edition. USA, New York: Manning Shelter Islands Publications Co. 538 p. ISBN: 978-1-61-729534-8.

16. BROY, Manfred, SIMMONS, Erik, BERENBACH, Brian, PAULISH J., Daniel, KAZMEIER, Juergen, RUDORFER, Arnold. 2009. Software & Systems Requirements Engineering: In Practice. USA, New York: The McGraw-Hill Companies, McGraw-Hill Professional. 326 p. ISBN: 978-0-07-160547-2.

17. FREEMAN, Adam. 2014. Pro Angular.js. Learn to harness the power of modern web browsers from within your application's code. USA, New York: Apress. 664 p. ISBN: 978-1-4302-6448-4.

18. RICHER, Justin, SANSO, Antonio, GLAZER, Ian. 2017. OAuth2 in action. USA, New York: Manning Shelter Islands Publications Co. 340 p. ISBN: 978-1-61-729327-6.

19. SOMMERVILLE, Ian. 1997. Software Engineering. Fifth Edition. England, Harlow: Addison Wesley Longman Limited. 752 p. ISBN: 0-201-427-656.

20. OTERO E., Carlos. 2012. Software Engineering Design. USA, Boca Raton, FL: Taylor & Francis Group, CRC Press. 346 p. ISBN: 978-750-8400.

21. REYNDERS, Fanie, HANSELMAN, Scott. 2018. Modern API Design with ASP.NET Core 2: Building Cross-Platform Back-End Systems. The Netherlands, Odijk: USA, New York: Apress. 244 p. ISBN: 978-1-4842-3518-8.

8. List of acronyms

SPA – single-page application

MPA – multiple-page application

IT – information technologies

NPM – Node Package Manager

RE – requirements engineering

ES – ECMAScript

JS – JavaScript

IDE – integrated development environment

SDLC – software development lifecycle

REST – Representational State Transfer

HTTP – Hypertext Transfer Protocol

UI – user interface

RMI – remote method invocation

UX – user experience

API – application programming interface

PHP – Personal Home Page, PHP: Hypertext Preprocessor

SSR – server-side rendering

SSG – static site generation

WebGL – web graphics library

TS – TypeScript

OS – operation system

V&V – validation & verification

SUT – system under test

MBT – model-based testing

SSR – server-side rendering

SSG – static site generation

Code sample

Both project structure and project code are presented below.

Project structure

GallerySPA – root folder.

The root folder contains 4 files:

1. public folder
2. src folder
3. babel.config.js
4. package.json

Public folder contains the following files:

- favicon.ico
- index.html

Src folder contains the following files:

- assets folder
- components folder
- services folder
- app.vue
- main.js

Assets folder contains:

- img folder

- the folder contains logo.gif and vue.gif

- logo.png file

Components folder contains:

- about.vue
- footer.vue
- header.vue
- picturesView.vue

Services folder contains:

- api.js

All the files mentioned are the elements of the gallery API.

The archive GallerySPA.7z contains all the files mentioned above.

Project code

1. **index.html file:**

```
<!DOCTYPE html>

<html lang="en-us">

 <head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <link rel="icon" href="<%= BASE_URL %> favicon.ico" />

  <title> <%= htmlWebpackPlugin.options.title %></title>

 </head>

 <body>

  <noscript> The project GallerySPA_web_axios <%=
htmlWebpackPlugin.options.title %> couldn't work properly.

   Please enable JavaScript in the browser to continue. </noscript>

  <div id="app"> </div>

 </body>

</html>
```

2. **about.vue file code:**

```
<template>
  <b-row class="About" id="about">
      <b-col md="6">
        <div class="textmodel">
            <h2> Some images </h2>
            <p> Photos are memories. They aims to help us to share our feeling
    with friends and family. The digital photo album makes it easy and lovely.
    Preserve the best moments with the help of the
    online gallery.</p>
        </div>
      </b-col>
      <b-col md="6">
        <div>
          <b-carousel id="carousel-no-animation" style="text-shadow: 0px 0px 2px #000"
no-animation controls indicators>
            <template v-for="(photo, index) in photos" >
              <b-carousel-slide
              :key="index"
              :img-src="photo.src.medium"
              img-width="400"
              style="height: 400px"
              ></b-carousel-slide>
            </template>
          </b-carousel>
        </div>
      </b-col>
    </b-row>
      <div style="margin-top:20px"><hr></div>
  </b-row>
</template>
<script>
export default {
  props:['photos']
}
</script>
```

3. **footer.vue file code:**

```
<template>

  <b-row>

    <b-col md="12">

      <p style="text-align: center; font-size: 14px"><b>Gallery SPA website</b></p>

    </b-col>

  </b-row>

</template>

<script>

export default {

}

</script>
```

4. header.vue file code:

```
<template>
  <b-row>
    <div class="header">
      <b-col lg="8" md="8" sm="6" xs="10">
          <img id="logo" src="../assets/img/logo.gif" alt="vueEmblem"/>
      </b-col>
      <b-col lg="4" md="4" sm="6" xs="2">
        <div class="navigation">
          <b-navbar toggleable="lg">
            <b-navbar-toggle target="navbar-toggle-collapse">
              <template #default="{ expanded }" >
                <b-icon v-if="expanded" icon="chevron-bar-up" ></b-icon>
                <b-icon v-else icon="chevron-bar-down"></b-icon>
              </template>
            </b-navbar-toggle>
            <b-collapse id="navbar-toggle-collapse" is-nav>
              <b-navbar-nav class="ml-auto">
                <b-nav-item href="#about"> ABOUT</b-nav-item>
                <b-nav-item href="#picturesView"> IMAGES</b-nav-item>
              </b-navbar-nav>
            </b-collapse>
          </b-navbar>
        </div>
      </b-col>
    </div>
  </b-row>
</template>
<script>
export default {
  created() {
    this.$root.$on('bv::scrollspy::activate', this.onActivate)
  },
  methods: {
    onActivate(target) {
      console.log('Received event: "bv::scrollspy::activate" for target ', target) } } }
</script>
```

5. **picturesView file code:**

```
<template>
   <b-row class="picturesview" id="view">
    <b-col md="12">
             <h1 class="titlepicturesview" id="picturesview"> GALLERY </h1>
      <hr id="maintitle">
    </b-col>
    <b-col md="12">
      <div class="textopicturesview">
        <h2 id="textpicturesview"> Embed your photos inside gallery </h2>
        <p id="textpicturesview">Show off your amazing photos with the gallery website! This is an easy<br/> and elegant way to share your memories with your family and friends all around the world! </p>
      </div>
    </b-col>
        <div id="view">
          <b-row>
            <b-col lg="12" xs="12" style="display:contents">
              <viewer :images="images" v-for="(photo, index) in photos" :key="index">
              <img :src="photo.src.medium" id="img" />
              </viewer>
      </b-col>

     </b-row>

    </div>

  <div style="margin-top:20px"><hr></div>

 </b-row>

</template>

<script>

export default {

 props:['photos'] }

</script>
```

6. api.js code sample:

```
import axios from 'axios'

const instance = axios.create({

  baseURL: 'https://api.pexels.com/v1/',

  headers: {

    Authorization:

  }

})

class ApiService {

  getPhotos(params = {}) {

    return instance.get('search', {

      params: params })}

  getPhotoById(id) {

    return instance.get('photos/' + id)  }}

export default new ApiService()
```

7. **main.js file code consists of:**

```
import Vue from 'vue'

import App from './App.vue'

import VueViewer from 'v-viewer'

import { BootstrapVue, IconsPlugin } from 'bootstrap-vue'

import 'viewerjs/dist/viewer.css'

Vue.use(VueViewer)

import 'bootstrap/dist/css/bootstrap.css'

import 'bootstrap-vue/dist/bootstrap-vue.css'

import { CarouselPlugin } from 'bootstrap-vue'

import { VBScrollspyPlugin } from 'bootstrap-vue'

Vue.use(VBScrollspyPlugin)

Vue.use(CarouselPlugin)

Vue.use(BootstrapVue)

Vue.use(IconsPlugin)

Vue.config.productionTip = false

new Vue({

  render: h => h(App)

}).$mount('#app')
```

8. babel.config.js code:

```
module.exports = {
  presets: [
    '@vue/cli-plugin-babel/preset'
  ]Some
}
```